HTML5 教程

HTML 5 简介

HTML5 是下一代的 HTML。

什么是 HTML5?

HTML5 将成为 HTML、XHTML 以及 HTML DOM 的新标准。

HTML 的上一个版本诞生于 1999 年。自从那以后，Web 世界已经经历了巨变。

HTML5 仍处于完善之中。然而，大部分现代浏览器已经具备了某些 HTML5 支持。

HTML5 是如何起步的？

HTML5 是 W3C 与 WHATWG 合作的结果。

编者注：W3C 指 World Wide Web Consortium，万维网联盟。

编者注：WHATWG 指 Web Hypertext Application Technology Working Group。

WHATWG 致力于 web 表单和应用程序，而 W3C 专注于 XHTML 2.0。在 2006 年，双方决定进行合作，来创建一个新版本的 HTML。

为 HTML5 建立的一些规则：

- 新特性应该基于 HTML、CSS、DOM 以及 JavaScript。
- 减少对外部插件的需求（比如 Flash）
- 更优秀的错误处理
- 更多取代脚本的标记
- HTML5 应该独立于设备
- 开发进程应对公众透明

新特性

HTML5 中的一些有趣的新特性：

- 用于绘画的 canvas 元素
- 用于媒介回放的 video 和 audio 元素
- 对本地离线存储的更好的支持
- 新的特殊内容元素，比如 article、footer、header、nav、section
- 新的表单控件，比如 calendar、date、time、email、url、search

浏览器支持

最新版本的 Safari、Chrome、Firefox 以及 Opera 支持某些 HTML5 特性。Internet Explorer 9 将支持某些 HTML5 特性。

HTML 5 视频

许多时髦的网站都提供视频。HTML5 提供了展示视频的标准。

Web 上的视频

直到现在，仍然不存在一项旨在网页上显示视频的标准。

今天，大多数视频是通过插件（比如 Flash）来显示的。然而，并非所有浏览器都拥有同样的插件。

HTML5 规定了一种通过 video 元素来包含视频的标准方法。

视频格式

当前，video 元素支持两种视频格式：

	Internet Explorer	Firefox 3.5	Opera 10.5	Chrome 3.0	Safari 3.0
Ogg		X	X	X	
MPEG 4				X	X

Ogg = 带有 Thedora 视频编码和 Vorbis 音频编码的 Ogg 文件
MPEG4 = 带有 H.264 视频编码和 AAC 音频编码的 MPEG 4 文件

如何工作

如需在 HTML5 中显示视频，您所有需要的是：

```
<video src="movie.ogg" controls="controls">
</video>
```

control 属性供添加播放、暂停和音量控件。

包含宽度和高度属性也是不错的主意。
<video> 与 </video> 之间插入的内容是供不支持 video 元素的浏览器显示的：

实例

```
<video src="movie.ogg" width="320" height="240" controls="controls">
Your browser does not support the video tag.
</video>
```

上面的例子使用一个 Ogg 文件，适用于 Firefox、Opera 以及 Chrome 浏览器。
要确保适用于 Safari 浏览器，视频文件必须是 MPEG4 类型。
video 元素允许多个 source 元素。source 元素可以链接不同的视频文件。浏览器将使用第一个可识别的格式：

实例

```
<video width="320" height="240" controls="controls">
  <source src="movie.ogg" type="video/ogg">
  <source src="movie.mp4" type="video/mp4">
Your browser does not support the video tag.
</video>
```

Internet Explorer

Internet Explorer 8 不支持 video 元素。在 IE 9 中，将提供对使用 MPEG4 的 video 元素的支持。

<video> 标签的属性

属性	值	描述
autoplay	autoplay	如果出现该属性，则视频在就绪后马上播放。
controls	controls	如果出现该属性，则向用户显示控件，比如播放按钮。
height	pixels	设置视频播放器的高度。
loop	loop	如果出现该属性，则当媒介文件完成播放后再次开始播放。
preload	preload	如果出现该属性，则视频在页面加载时进行加载，并预备播放。 如果使用 "autoplay"，则忽略该属性。
src	url	要播放的视频的 URL。
width	pixels	设置视频播放器的宽度。

HTML 5 音频

HTML5 提供了播放音频的标准。

Web 上的音频

直到现在，仍然不存在一项旨在网页上播放音频的标准。
今天，大多数音频是通过插件（比如 Flash）来播放的。然而，并非所有浏览器都拥有同样的插件。
HTML5 规定了一种通过 audio 元素来包含音频的标准方法。
audio 元素能够播放声音文件或者音频流。

视频格式

当前，audio 元素支持三种音频格式：

	Internet Explorer	Firefox 3.5	Opera 10.5	Chrome 3.0	Safari 3.0
Ogg Vorbis		X	X	X	
MP3				X	X
Wav		X	X		X

如何工作

如需在 HTML5 中播放音频，您所有需要的是：

```
<audio src="song.ogg" controls="controls">

</audio>
```

control 属性供添加播放、暂停和音量控件。
<audio> 与 </audio> 之间插入的内容是供不支持 audio 元素的浏览器显示的：
实例

```
<audio src="song.ogg" controls="controls">
Your browser does not support the audio tag.
</audio>
```

上面的例子使用一个 Ogg 文件，适用于 Firefox、Opera 以及 Chrome 浏览器。
要确保适用于 Safari 浏览器，音频文件必须是 MP3 或 Wav 类型。
audio 元素允许多个 source 元素。source 元素可以链接不同的音频文件。浏览器将使用第一个可识别的格式：
实例

```
<audio controls="controls">
   <source src="song.ogg" type="audio/ogg">
   <source src="song.mp3" type="audio/mpeg">
Your browser does not support the audio tag.
</audio>
```

Internet Explorer

Internet Explorer 8 不支持 audio 元素。在 IE 9 中，将提供对 audio 元素的支持。
<audio> 标签的属性

属性	值	描述
autoplay	autoplay	如果出现该属性，则音频在就绪后马上播放。
controls	controls	如果出现该属性，则向用户显示控件，比如播放按钮。
preload	preload	如果出现该属性，则音频在页面加载时进行加载，并预备播放。 如果使用 "autoplay"，则忽略该属性。
src	url	要播放的音频的 URL。

HTML 5 Canvas

canvas 元素用于在网页上绘制图形。

什么是 Canvas?

HTML5 的 canvas 元素使用 JavaScript 在网页上绘制图像。
画布是一个矩形区域，您可以控制其每一像素。
canvas 拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。

创建 Canvas 元素

向 HTML5 页面添加 canvas 元素。
规定元素的 id、宽度和高度：

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

通过 JavaScript 来绘制

canvas 元素本身是没有绘图能力的。所有的绘制工作必须在 JavaScript 内部完成：

```
<script type="text/javascript">
var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.fillStyle="#FF0000";
cxt.fillRect(0,0,150,75);
</script>
```

JavaScript 使用 id 来寻找 canvas 元素：

```
var c=document.getElementById("myCanvas");
```

然后，创建 context 对象：

```
var cxt=c.getContext("2d");
```

getContext("2d") 对象是内建的 HTML5 对象，拥有多种绘制路径、矩形、圆形、字符以及

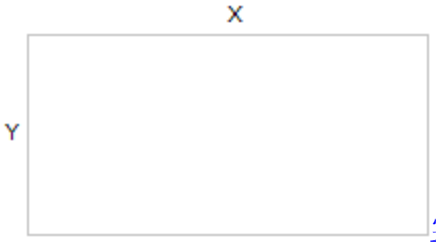
添加图像的方法。
下面的两行代码绘制一个红色的矩形：

```
cxt.fillStyle="#FF0000";
cxt.fillRect(0,0,150,75);
```

fillStyle 方法将其染成红色，fillRect 方法规定了形状、位置和尺寸。

理解坐标

上面的 fillRect 方法拥有参数 (0,0,150,75)。
意思是：在画布上绘制 150x75 的矩形，从左上角开始 (0,0)。
如下图所示，画布的 X 和 Y 坐标用于在画布上对绘画进行定位。



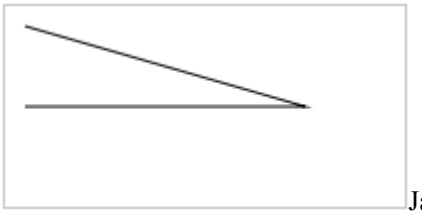
实 例：把鼠标悬停在矩形上可以看到坐标

更多 Canvas 实例

下面的在 canvas 元素上进行绘画的更多实例：

实例 - 线条

通过指定从何处开始，在何处结束，来绘制一条线：


JavaScript 代码：

```
<script type="text/javascript">

var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.moveTo(10,10);
cxt.lineTo(150,50);
cxt.lineTo(10,50);
cxt.stroke();

</script>
```

canvas 元素：

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
Your browser does not support the canvas element.
</canvas>
```

亲自 试一试

实例 - 圆形

通过规定尺寸、颜色和位置，来绘制一个圆：

JavaScript 代码：

```
<script type="text/javascript">

var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
cxt.fillStyle="#FF0000";
cxt.beginPath();
cxt.arc(70,18,15,0,Math.PI*2,true);
cxt.closePath();
cxt.fill();

</script>
```

canvas 元素：

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
Your browser does not support the canvas element.
</canvas>
```

亲 自试一试

实例 - 渐变

使用您指定的颜色来绘制渐变背景：

JavaScript 代码：

```
<script type="text/javascript">

var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
var grd=cxt.createLinearGradient(0,0,175,50);
grd.addColorStop(0,"#FF0000");
grd.addColorStop(1,"#00FF00");
cxt.fillStyle=grd;
cxt.fillRect(0,0,175,50);

</script>
```

canvas 元素：

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
Your browser does not support the canvas element.
</canvas>
```

亲自试一 试

实例 - 图像

把一幅图像放置到画布上：

JavaScript 代码：

```
<script type="text/javascript">

var c=document.getElementById("myCanvas");
var cxt=c.getContext("2d");
var img=new Image()
img.src="flower.png"
cxt.drawImage(img,0,0);

</script>
```

canvas 元素：

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
Your browser does not support the canvas element.
</canvas>
```

HTML 5 Web 存储

在客户端存储数据

HTML5 提供了两种在客户端存储数据的新方法：
localStorage - 没有时间限制的数据存储
sessionStorage - 针对一个 session 的数据存储
之前，这些都是由 cookie 完成的。但是 cookie 不适合大量数据的存储，因为它们由每个对服务器的请求来传递，这使得 cookie 速度很慢而且效率也不高。
在 HTML5 中，数据不是由每个服务器请求传递的，而是只有在请求时使用数据。它使在不影响网站性能的情况下存储大量数据成为可能。
对于不同的网站，数据存储于不同的区域，并且一个网站只能访问其自身的数据。
HTML5 使用 JavaScript 来存储和访问数据。

localStorage 方法

localStorage 方法存储的数据没有时间限制。第二天、第二周或下一年之后，数据依然可用。
如何创建和访问 localStorage：

实例

```
<script type="text/javascript">
localStorage.lastname="Smith";
document.write(localStorage.lastname);
</script>
```

下面的例子对用户访问页面的次数进行计数：

实例

```
<script type="text/javascript">
if (localStorage.pagecount)
  {
  localStorage.pagecount=Number(localStorage.pagecount) +1;
  }
else
  {
  localStorage.pagecount=1;
  }
document.write("Visits "+ localStorage.pagecount + " time(s).");
</script>
```

sessionStorage 方法

sessionStorage 方法针对一个 session 进行数据存储。当用户关闭浏览器窗口后，数据会被

删除。
如何创建并访问一个 sessionStorage：

实例

```
<script type="text/javascript">
sessionStorage.lastname="Smith";
document.write(sessionStorage.lastname);
</script>
```

下面的例子对用户在当前 session 中访问页面的次数进行计数：

实例

```
<script type="text/javascript">
if (sessionStorage.pagecount)
  {
  sessionStorage.pagecount=Number(sessionStorage.pagecount) +1;
  }
else
  {
  sessionStorage.pagecount=1;
  }
document.write("Visits "+sessionStorage.pagecount+" time(s) this session.");
</script>
```

HTML5 Input 类型

HTML5 新的 Input 类型

HTML5 拥有多个新的表单输入类型。这些新特性提供了更好的输入控制和验证。
本章全面介绍这些新的输入类型：
email
url
number
range
Date pickers (date, month, week, time, datetime, datetime-local)
search
color

浏览器支持

Input type	IE	Firefox	Opera	Chrome	Safari
email	No	No	9.0	No	No

url	No	No	9.0	No	No
number	No	No	9.0	No	No
range	No	No	9.0	4.0	4.0
Date pickers	No	No	9.0	No	No
search	No	No	No	No	No
color	No	No	No	No	No

注释：Opera 对新的输入类型的支持最好。不过您已经可以在所有主流的浏览器中使用它们了。即使不被支持，仍然可以显示为常规的文本域。

Input 类型 - email

email 类型用于应该包含 e-mail 地址的输入域。
在提交表单时，会自动验证 email 域的值。

实例

E-mail: <input type="email" name="user_email" />
提示：iPhone 中的 Safari 浏览器支持 email 输入类型，并通过改变触摸屏键盘来配合它（添加 @ 和 .com 选项）。

Input 类型 - url

url 类型用于应该包含 URL 地址的输入域。
在提交表单时，会自动验证 url 域的值。

实例

Homepage: <input type="url" name="user_url" />
提示：iPhone 中的 Safari 浏览器支持 url 输入类型，并通过改变触摸屏键盘来配合它（添加 .com 选项）。

Input 类型 - number

number 类型用于应该包含数值的输入域。
您还能够设定对所接受的数字的限定：

实例

Points: <input type="number" name="points" min="1" max="10" />

请使用下面的属性来规定对数字类型的限定：

属性	值	描述
max	number	规定允许的最大值
min	number	规定允许的最小值
step	number	规定合法的数字间隔（如果 step="3"，则合法的数是 -3,0,3,6 等）
value	number	规定默认值

提示：iPhone 中的 Safari 浏览器支持 number 输入类型，并通过改变触摸屏键盘来配合它（显示数字）。

Input 类型 - range

range 类型用于应该包含一定范围内数字值的输入域。
range 类型显示为滑动条。
您还能够设定对所接受的数字的限定：

实例

<input type="range" name="points" min="1" max="10" />
请使用下面的属性来规定对数字类型的限定：

属性	值	描述
max	number	规定允许的最大值
min	number	规定允许的最小值
step	number	规定合法的数字间隔（如果 step="3"，则合法的数是 -3,0,3,6 等）
value	number	规定默认值

Input 类型 - Date Pickers（数据检出器）

HTML5 拥有多个可供选取日期和时间的新输入类型：
date - 选取日、月、年
month - 选取月、年
week - 选取周和年
time - 选取时间（小时和分钟）
datetime - 选取时间、日、月、年（UTC 时间）
datetime-local - 选取时间、日、月、年（本地时间）
下面的例子允许您从日历中选取一个日期：

实例

Date: <input type="date" name="user_date" />
输入类型 "month":
输入类型 "week":
输入类型 "time":
输入类型 "datetime":
输入类型 "datetime-local":

Input 类型 - search

search 类型用于搜索域，比如站点搜索或 Google 搜索。
search 域显示为常规的文本域。

HTML5 表单元素

HTML5 的新的表单元素：

HTML5 拥有若干涉及表单的元素和属性。
本章介绍以下新的表单元素：
datalist
keygen
output

浏览器支持

Input type	IE	Firefox	Opera	Chrome	Safari
datalist	No	No	9.5	No	No
keygen	No	No	10.5	3.0	No
output	No	No	9.5	No	No

datalist 元素

datalist 元素规定输入域的选项列表。
列表是通过 datalist 内的 option 元素创建的。
如需把 datalist 绑定到输入域，请用输入域的 list 属性引用 datalist 的 id：

实例

Webpage: <input type="url" list="url_list" name="link" />
<datalist id="url_list">
<option label="W3School" value="http://www.W3School.com.cn" />
<option label="Google" value="http://www.google.com" />
<option label="Microsoft" value="http://www.microsoft.com" />
</datalist>
提示：option 元素永远都要设置 value 属性。

keygen 元素

keygen 元素的作用是提供一种验证用户的可靠方法。
keygen 元素是密钥对生成器（key-pair generator）。当提交表单时，会生成两个键，一个是私钥，一个公钥。
私钥（private key）存储于客户端，公钥（public key）则被发送到服务器。公钥可用于之后验证用户的客户端证书（client certificate）。
目前，浏览器对此元素的糟糕的支持度不足以使其成为一种有用的安全标准。

实例

```
<form action="demo_form.asp" method="get">
Username: <input type="text" name="usr_name" />
Encryption: <keygen name="security" />
<input type="submit" />
</form>
```

output 元素

output 元素用于不同类型的输出，比如计算或脚本输出：

实例

```
<output id="result" onforminput="resCalc()"></output>
```

HTML5 表单属性

HTML5 的新的表单属性

本章讲解涉及 <form> 和 <input> 元素的新属性。

新的 form 属性：

- autocomplete
- novalidate

新的 input 属性：

- autocomplete
- autofocus
- form
- form overrides (formaction, formenctype, formmethod, formnovalidate, formtarget)
- height 和 width
- list
- min, max 和 step
- multiple
- pattern (regexp)
- placeholder
- required

浏览器支持

Input type	IE	Firefox	Opera	Chrome	Safari
autocomplete	8.0	3.5	9.5	3.0	4.0

autofocus	No	No	10.0	3.0	4.0
form	No	No	9.5	No	No
form overrides	No	No	10.5	No	No
height and width	8.0	3.5	9.5	3.0	4.0
list	No	No	9.5	No	No
min, max and step	No	No	9.5	3.0	No
multiple	No	3.5	No	3.0	4.0
novalidate	No	No	No	No	No
pattern	No	No	9.5	3.0	No
placeholder	No	No	No	3.0	3.0
required	No	No	9.5	3.0	No

autocomplete 属性

autocomplete 属性规定 form 或 input 域应该拥有自动完成功能。
注释：autocomplete 适用于 <form> 标签，以及以下类型的 <input> 标签：text, search, url, telephone, email, password, datepickers, range 以及 color。
当用户在自动完成域中开始输入时，浏览器应该在该域中显示填写的选项：

实例

```
<form action="demo_form.asp" method="get" autocomplete="on">
First name: <input type="text" name="fname" /><br />
Last name: <input type="text" name="lname" /><br />
E-mail: <input type="email" name="email" autocomplete="off" /><br />
<input type="submit" />
</form>
```

注释：在某些浏览器中，您可能需要启用自动完成功能，以使该属性生效。

autofocus 属性

autofocus 属性规定在页面加载时，域自动地获得焦点。
注释：autofocus 属性适用于所有 <input> 标签的类型。

实例

```
User name: <input type="text" name="user_name"    autofocus="autofocus" />
```

form 属性

form 属性规定输入域所属的一个或多个表单。
注释：form 属性适用于所有 <input> 标签的类型。
form 属性必须引用所属表单的 id：

实例

```
<form action="demo_form.asp" method="get" id="user_form">
First name:<input type="text" name="fname" />
<input type="submit" />
</form>
Last name: <input type="text" name="lname" form="user_form" />
```
注释：如需引用一个以上的表单，请使用空格分隔的列表。

表单重写属性

表单重写属性（form override attributes）允许您重写 form 元素的某些属性设定。
表单重写属性有：
formaction - 重写表单的 action 属性
formenctype - 重写表单的 enctype 属性
formmethod - 重写表单的 method 属性
formnovalidate - 重写表单的 novalidate 属性
formtarget - 重写表单的 target 属性
注释：表单重写属性适用于以下类型的 <input> 标签：submit 和 image。

实例

```
<form action="demo_form.asp" method="get" id="user_form">
E-mail: <input type="email" name="userid" /><br />
<input type="submit" value="Submit" />
<br />
<input type="submit" formaction="demo_admin.asp" value="Submit as admin" />
<br />
<input type="submit" formnovalidate="true" value="Submit without validation" />
<br />
</form>
```
注释：这些属性对于创建不同的提交按钮很有帮助。

height 和 width 属性

height 和 width 属性规定用于 image 类型的 input 标签的图像高度和宽度。
注释：height 和 width 属性只适用于 image 类型的 <input> 标签。

实例

```
<input type="image" src="img_submit.gif" width="99" height="99" />
```

list 属性

list 属性规定输入域的 datalist。datalist 是输入域的选项列表。
注释：list 属性适用于以下类型的 <input> 标签：text, search, url, telephone, email, date pickers, number, range 以及 color。

实例

Webpage: <input type="url" list="url_list" name="link" />
<datalist id="url_list">
<option label="W3Schools" value="http://www.w3school.com.cn" />
<option label="Google" value="http://www.google.com" />
<option label="Microsoft" value="http://www.microsoft.com" />
</datalist>

min、max 和 step 属性

min、max 和 step 属性用于为包含数字或日期的 input 类型规定限定（约束）。
max 属性规定输入域所允许的最大值。
min 属性规定输入域所允许的最小值。
step 属性为输入域规定合法的数字间隔（如果 step="3"，则合法的数是 -3,0,3,6 等）。
注释：min、max 和 step 属性适用于以下类型的 <input> 标签：date pickers、number 以及 range。
下面的例子显示一个数字域，该域接受介于 0 到 10 之间的值，且步进为 3（即合法的值为 0、3、6 和 9）：

实例

Points: <input type="number" name="points" min="0" max="10" step="3" />

multiple 属性

multiple 属性规定输入域中可选择多个值。
注释：multiple 属性适用于以下类型的 <input> 标签：email 和 file。

实例

Select images: <input type="file" name="img" multiple="multiple" />

novalidate 属性

novalidate 属性规定在提交表单时不应该验证 form 或 input 域。
注释：novalidate 属性适用于 <form> 以及以下类型的 <input> 标签：text, search, url, telephone, email, password, date pickers, range 以及 color.

实例

<form action="demo_form.asp" method="get" novalidate="true">
E-mail: <input type="email" name="user_email" />
<input type="submit" />
</form>

pattern 属性

pattern 属性规定用于验证 input 域的模式（pattern）。
模式（pattern） 是正则表达式。您可以在我们的 JavaScript 教程中学习到有关正则 表达式的内容。
注释：pattern 属性适用于以下类型的 <input> 标签：text, search, url, telephone, email 以及 password。
下面的例子显示了一个只能包含三个字母的文本域（不含数字及特殊字符）：

实例

```
Country code: <input type="text" name="country_code"
pattern="[A-z]{3}" title="Three letter country code" />
```

placeholder 属性

placeholder 属性提供一种提示（hint），描述输入域所期待的值。
注释：placeholder 属性适用于以下类型的 <input> 标签：text, search, url, telephone, email 以及 password。
提示（hint）会在输入域为空时显示出现，会在输入域获得焦点时消失：

实例

```
<input type="search" name="user_search"    placeholder="Search W3School" />
```

required 属性

required 属性规定必须在提交之前填写输入域（不能为空）。
注释：required 属性适用于以下类型的 <input> 标签：text, search, url, telephone, email, password, date pickers, number, checkbox, radio 以及 file。

实例

```
Name: <input type="text" name="usr_name" required="required" />
```

HTML 5 参考手册

W3C 在 1 月 22 日发布了最新的 HTML 5 工作草案。HTML 5 工作组包括 AOL, Apple, Google, IBM, Microsoft, Mozilla, Nokia, Opera 以及数百个其他的开发商。HTML 5 中的一些新特性：嵌入音频、视频、图片的函数、 客户端数据存储，以及交互式文档。其他特性包括新的页面元素，比如 <header>, <section>, <footer>, 以及 <figure>。
通过制定如何处理所有 HTML 元素以及如何从错误中恢复的精确规则，HTML 5 改进了互操作性，并减少了开发成本。

按字母顺序排列

- *4*: 指示在 HTML 4.01 中是否定义了该元素
- *5*: 指示在 HTML 5 中是否定义了该元素

标签	描述	4	5
<!--...-->	定义注释。	4	5
<!DOCTYPE>	定义文档类型。	4	5
<a>	定义超链接。	4	5
<abbr>	定义缩写。	4	5
<acronym>	HTML 5 中不支持。定义首字母缩写。	4	
<address>	定义地址元素。	4	5
<applet>	HTML 5 中不支持。定义 applet。	4	
<area>	定义图像映射中的区域。	4	5
<article>	定义 article。		5
<aside>	定义页面内容之外的内容。		5
<audio>	定义声音内容。		5
<b>	定义粗体文本。	4	5
<base>	定义页面中所有链接的基准 URL。	4	5
<basefont>	HTML 5 中不支持。请使用 CSS 代替。	4	
<bdo>	定义文本显示的方向。	4	5
<big>	HTML 5 中不支持。定义大号文本。	4	
<blockquote>	定义长的引用。	4	5
<body>	定义 body 元素。	4	5
 	插入换行符。	4	5
<button>	定义按钮。	4	5
<canvas>	定义图形。		5
<caption>	定义表格标题。	4	5
	HTML 5 中不支持。定义居中的文本。	4	
<cite>	定义引用。	4	5
<code>	定义计算机代码文本。	4	5
<col>	定义表格列的属性。	4	5
<colgroup>	定义表格列的分组。	4	5
<command>	定义命令按钮。		5
<datalist>	定义下拉列表。		5
<dd>	定义定义的描述。	4	5
<del>	定义删除文本。	4	5
<details>	定义元素的细节。		5
<dfn>	定义定义项目。	4	5
<dir>	HTML 5 中不支持。定义目录列表。	4	
	定义文档中的一个部分。	4	5
<dl>	定义定义列表。	4	5
<dt>	定义定义的项目。	4	5
	定义强调文本。	4	5

<embed>	定义外部交互内容或插件。		5
<fieldset>	定义 fieldset。	4	5
<figcaption>	定义 figure 元素的标题。		5
<figure>	定义媒介内容的分组，以及它们的标题。		5
<font>	HTML 5 中不支持。	4	
<footer>	定义 section 或 page 的页脚。		5
<form>	定义表单。	4	5
<frame>	HTML 5 中不支持。定义子窗口（框架）。	4	
<frameset>	HTML 5 中不支持。定义框架的集。	4	
<h1> to <h6>	定义标题 1 到标题 6。	4	5
<head>	定义关于文档的信息。	4	5
<header>	定义 section 或 page 的页眉。		5
<hgroup>	定义有关文档中的 section 的信息。		5
<hr>	定义水平线。	4	5
<html>	定义 html 文档。	4	5
<i>	定义斜体文本。	4	5
<iframe>	定义行内的子窗口（框架）。	4	5
<img>	定义图像。	4	5
<input>	定义输入域。	4	5
<ins>	定义插入文本。	4	5
<keygen>	定义生成密钥。		5
<isindex>	HTML 5 中不支持。定义单行的输入域。	4	
<kbd>	定义键盘文本。	4	5
<label>	定义表单控件的标注。	4	5
<legend>	定义 fieldset 中的标题。	4	5
	定义列表的项目。	4	5
<link>	定义资源引用。	4	5
<map>	定义图像映射。	4	5
<mark>	定义有记号的文本。		5
<menu>	定义菜单列表。	4	5
<meta>	定义元信息。	4	5
<meter>	定义预定义范围内的度量。		5
<nav>	定义导航链接。		5
<noframes>	HTML 5 中不支持。定义 noframe 部分。	4	
<noscript>	定义 noscript 部分。	4	5
<object>	定义嵌入对象。	4	5
	定义有序列表。	4	5
<optgroup>	定义选项组。	4	5
<option>	定义下拉列表中的选项。	4	5
<output>	定义输出的一些类型。		5
<p>	定义段落。	4	5
<param>	为对象定义参数。	4	5
<pre>	定义预格式化文本。	4	5
<progress>	定义任何类型的任务的进度。		5

<q>	定义短的引用。	4	5
<rp>	定义若浏览器不支持 ruby 元素显示的内容。		5
<rt>	定义 ruby 注释的解释。		5
<ruby>	定义 ruby 注释。		5
<s>	HTML 5 中不支持。定义加删除线的文本。	4	
<samp>	定义样本计算机代码。	4	5
<script>	定义脚本。	4	5
<section>	定义 section。		5
<select>	定义可选列表。	4	5
<small>	定义小号文本。	4	5
<source>	定义媒介源。		5
	定义文档中的 section。	4	5
<strike>	HTML 5 中不支持。定义加删除线的文本。	4	
	定义强调文本。	4	5
<style>	定义样式定义。	4	5
<sub>	定义下标文本。	4	5
<summary>	定义 details 元素的标题。		5
<sup>	定义上标文本。	4	5
<table>	定义表格。	4	5
<tbody>	定义表格的主体。	4	5
<td>	定义表格单元。	4	5
<textarea>	定义 textarea。	4	5
<tfoot>	定义表格的脚注。	4	5
<th>	定义表头。	4	5
<thead>	定义表头。	4	5
<time>	定义日期/时间。		5
<title>	定义文档的标题。	4	5
<tr>	定义表格行。	4	5
<tt>	HTML 5 中不支持。定义打字机文本。	4	
<u>	HTML 5 中不支持。定义下划线文本。	4	
	定义无序列表。	4	5
<var>	定义变量。	4	5
<video>	定义视频。		5
<xmp>	HTML 5 中不支持。定义预格式文本。	4	

HTML 5 标准属性

所有 HTML 5 标签均支持下面列出的属性，仅有少数例外。

HTML 5 标准属性

NEW：HTML 5 中新的标准属性。
注释：HTML 4.01 不再支持 accesskey 属性：

属性	值	描述
accesskey	*character*	规定访问元素的键盘快捷键
class	*classname*	规定元素的类名（用于规定样式表中的类）。
contenteditable	• true • false	规定是否允许用户编辑内容。
contextmenu	*menu_id*	规定元素的上下文菜单。
data-yourvalue	*value*	创作者定义的属性。 HTML 文档的创作者可以定义他们自己的属性。 必须以 "data-" 开头。
dir	• ltr • rtl	规定元素中内容的文本方向。
draggable	• true • false • auto	规定是否允许用户拖动元素。
hidden	hidden	规定该元素是无关的。被隐藏的元素不会显示。
id	*id*	规定元素的唯一 ID。
item	• *empty* • *url*	用于组合元素。
itemprop	• *url* • *group* *value*	用于组合项目。
lang	*language_code*	规定元素中内容的语言代码。语言代码参考手册。
spellcheck	• true • false	规定是否必须对元素进行拼写或语法检查。
style	*style_definition*	规定元素的行内样式。
subject	*id*	规定元素对应的项目。
tabindex	*number*	规定元素的 tab 键控制次序。
title	*text*	规定有关元素的额外信息。

HTML 5 事件属性

标准事件属性

HTML 4 增加了通过事件触发浏览器中行为的能力，比如当用户点击某个元素时启动一段 JavaScript。

下面的表格列出了可插入 HTML 5 元素中以定义事件行为的标准事件属性。

Window 事件属性

window 对象触发的事件。
适用于 <body> 标签：

属性	值	描述
onafterprint	*script*	在打印文档之后运行脚本
onbeforeprint	*script*	在文档打印之前运行脚本
onbeforeonload	*script*	在文档加载之前运行脚本
onblur	*script*	当窗口失去焦点时运行脚本
onerror	*script*	当错误发生时运行脚本
onfocus	*script*	当窗口获得焦点时运行脚本
onhaschange	*script*	当文档改变时运行脚本
onload	*script*	当文档加载时运行脚本
onmessage	*script*	当触发消息时运行脚本
onoffline	*script*	当文档离线时运行脚本
ononline	*script*	当文档上线时运行脚本
onpagehide	*script*	当窗口隐藏时运行脚本
onpageshow	*script*	当窗口可见时运行脚本
onpopstate	*script*	当窗口历史记录改变时运行脚本
onredo	*script*	当文档执行再执行操作（redo）时运行脚本
onresize	*script*	当调整窗口大小时运行脚本
onstorage	*script*	当文档加载加载时运行脚本
onundo	*script*	当文档执行撤销操作时运行脚本
onunload	*script*	当用户离开文档时运行脚本

表单事件

由 HTML 表单内部的动作触发的事件。
适用于所有 HTML 5 元素，不过最常用于表单元素中：

属性	值	描述
onblur	*script*	当元素失去焦点时运行脚本
onchange	*script*	当元素改变时运行脚本
oncontextmenu	*script*	当触发上下文菜单时运行脚本

onfocus	*script*	当元素获得焦点时运行脚本
onformchange	*script*	当表单改变时运行脚本
onforminput	*script*	当表单获得用户输入时运行脚本
oninput	*script*	当元素获得用户输入时运行脚本
oninvalid	*script*	当元素无效时运行脚本
onreset	*script*	当表单重置时运行脚本。HTML 5 不支持。
onselect	*script*	当选取元素时运行脚本
onsubmit	*script*	当提交表单时运行脚本

键盘事件

由键盘触发的事件。
适用于所有 HTML 5 元素：

属性	值	描述
onkeydown	*script*	当按下按键时运行脚本
onkeypress	*script*	当按下并松开按键时运行脚本
onkeyup	*script*	当松开按键时运行脚本

鼠标事件

由鼠标货相似的用户动作触发的事件。
适用于所有 HTML 5 元素：

属性	值	描述
onclick	*script*	当单击鼠标时运行脚本
ondblclick	*script*	当双击鼠标时运行脚本
ondrag	*script*	当拖动元素时运行脚本
ondragend	*script*	当拖动操作结束时运行脚本
ondragenter	*script*	当元素被拖动至有效的拖放目标时运行脚本
ondragleave	*script*	当元素离开有效拖放目标时运行脚本
ondragover	*script*	当元素被拖动至有效拖放目标上方时运行脚本
ondragstart	*script*	当拖动操作开始时运行脚本
ondrop	*script*	当被拖动元素正在被拖放时运行脚本
onmousedown	*script*	当按下鼠标按钮时运行脚本
onmousemove	*script*	当鼠标指针移动时运行脚本
onmouseout	*script*	当鼠标指针移出元素时运行脚本
onmouseover	*script*	当鼠标指针移至元素之上时运行脚本
onmouseup	*script*	当松开鼠标按钮时运行脚本
onmousewheel	*script*	当转动鼠标滚轮时运行脚本
onscroll	*script*	当滚动元素滚动元素的滚动条时运行脚本

媒介事件

由视频、图像以及音频等媒介触发的事件。
适用于所有 HTML 5 元素，不过在媒介元素（诸如 audio、embed、img、object 以及 video）

中最常用：

属性	值	描述
onabort	*script*	当发生中指事件时运行脚本
oncanplay	*script*	当媒介能够开始播放但可能因缓冲而需要停止时运行脚本
oncanplaythrough	*script*	当媒介能够无需因缓冲而停止即可播放至结尾时运行脚本
ondurationchange	*script*	当媒介长度改变时运行脚本
onemptied	*script*	当媒介资源元素突然为空时（网络错误、加载错误等）运行脚本
onended	*script*	当媒介已抵达结尾时运行脚本
onerror	*script*	当在元素加载期间发生错误时运行脚本
onloadeddata	*script*	当加载媒介数据时运行脚本
onloadedmetadata	*script*	当媒介元素的持续时间以及其他媒介数据已加载时运行脚本
onloadstart	*script*	当浏览器开始加载媒介数据时运行脚本
onpause	*script*	当媒介数据暂停时运行脚本
onplay	*script*	当媒介数据将要开始播放时运行脚本
onplaying	*script*	当媒介数据已开始播放时运行脚本
onprogress	*script*	当浏览器正在取媒介数据时运行脚本
onratechange	*script*	当媒介数据的播放速率改变时运行脚本
onreadystatechange	*script*	当就绪状态（ready-state）改变时运行脚本
onseeked	*script*	当媒介元素的定位属性 [1] 不再为真且定位已结束时运行脚本
onseeking	*script*	当媒介元素的定位属性为真且定位已开始时运行脚本
onstalled	*script*	当取回媒介数据过程中（延迟）存在错误时运行脚本
onsuspend	*script*	当浏览器已在取媒介数据但在取回整个媒介文件之前停止时运行脚本
ontimeupdate	*script*	当媒介改变其播放位置时运行脚本

onvolumechange	*script*	当媒介改变音量亦或当音量被设置为静音时运行脚本
onwaiting	*script*	当媒介已停止播放但打算继续播放时运行脚本

[1]：定位属性的英文译文是：seeking attribute。